|  |  |  |
| --- | --- | --- |
| A képen szöveg, Betűtípus, embléma, szimbólum látható  Automatikusan generált leírás | **MISKOLCI EGYETEM**  **Gépészmérnöki és Informatikai Kar**  **3515 Miskolc-Egyetemváros** | C:\Users\Krisztina\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\50640BF4.tmp |

**Mobilprogramozás**

**Beadandó**

**program**

**dokumentáció**

Készítette: Danyi Károly

Neptunkód: RDC1L4

Miskolc, 2024.

Tartalom

[1. Bevezetés 3](#_Toc182920021)

[A beadandó célja egy React Native-alapú mobilalkalmazás fejlesztése, amely a Firebase-t használja a felhasználói hitelesítés és adatkezelés támogatására. Az alkalmazás fő funkciói közé tartozik a felhasználói fiókok kezelése, a napi bejegyzések rögzítése (jegyzetek hozzáadása adott napokhoz), valamint a profilbeállítások testreszabása. Pl.: Jelsz 3](#_Toc182920022)

[2. Funkciók 4](#_Toc182920023)

[2.1. Felhasználói Hitelesítés 4](#_Toc182920024)

[2.2. Adatkezelés és Teendő Menedzsment 4](#_Toc182920025)

[2.3. Napi Teendők Megjelenítése 5](#_Toc182920026)

[2.4. Navigáció 5](#_Toc182920027)

[2.5. Profil Kezelés 5](#_Toc182920028)

[2.6. Adatmegőrzés és Szinkronizálás 6](#_Toc182920029)

[3. Használt Fejlesztési és Tesztelési Környezetek 7](#_Toc182920030)

[3.1. Fejlesztési Környezet: Visual Studio Code 7](#_Toc182920031)

[3.2. Tesztelési Környezet: Android Studio 8](#_Toc182920032)

[4. Kód 9](#_Toc182920033)

[4.1 Firebase Konfiguráció 9](#_Toc182920034)

[4.2 Navigáció 10](#_Toc182920035)

[4.2.1 Main Nav. 10](#_Toc182920036)

[4.2.2 Bottom Nav. 10](#_Toc182920037)

[4.3 Komponensek 11](#_Toc182920038)

[4.3.1 AuthScreen 11](#_Toc182920039)

[4.3.2 DataSet 11](#_Toc182920040)

[4.3.3 ProfileScreen 11](#_Toc182920041)

[4.3.4 DayScreen 11](#_Toc182920042)

[4.3.5 MainTabs 11](#_Toc182920043)

[4.4. Funkciók Részletezése 11](#_Toc182920044)

[4.4.1 Felhasználói Hitelesítés: 11](#_Toc182920045)

[4.4.2 Adatkezelés és Mentés 11](#_Toc182920046)

[4.4.3 Napi Értesítések: 11](#_Toc182920047)

[4.5 Stílusok 13](#_Toc182920048)

# 1. Bevezetés

A képen szöveg, képernyőkép, Multimédiás szoftver, Grafikai szoftver látható

Automatikusan generált leírásA képen szöveg, képernyőkép, Multimédiás szoftver, Grafikai szoftver látható

Automatikusan generált leírásA beadandó célja egy React Native-alapú mobilalkalmazás fejlesztése, amely a Firebase-t használja a felhasználói hitelesítés és adatkezelés támogatására. Az alkalmazás fő funkciói közé tartozik a felhasználói fiókok kezelése, a napi bejegyzések rögzítése (jegyzetek hozzáadása adott napokhoz), valamint a profilbeállítások testreszabása. Pl.: Jelsz ó módosítás

# 2. Funkciók

2.1. Felhasználói Hitelesítés

Bejelentkezés és regisztráció: Az alkalmazás lehetővé teszi a felhasználók számára, hogy regisztráljanak és bejelentkezzenek. Az autentikáció a Firebase segítségével történik, amely e-mail alapú hitelesítést alkalmaz. Új jelszó beállítása: A felhasználók jelszót módosíthatnak a profil oldalon keresztül. A jelszó módosítását követően a felhasználó automatikusan ki van jelentkeztetve, és visszakerül a kezdőképernyőre. Kilépés: A felhasználók bármikor kijelentkezhetnek az alkalmazásból, amely a Firebase signOut funkcióját használja.

2.2. Adatkezelés és Teendő Menedzsment

Teendők hozzáadása és mentése: A felhasználók napi teendőket adhatnak hozzá, amelyeket napra lebontva menthetnek el. A teendők tartalmazzák a napot és egy szabad szöveges leírást. Teendők mentése: A felhasználók által hozzáadott napi teendők elmentésre kerülnek, és megjelennek a választott naphoz kapcsolódóan. Az alkalmazás figyelmezteti a felhasználót, ha a leírás mező üres. Felhasználói interakció: A teendőkhez checkbox is tartozik, amellyel a felhasználó jelezheti, ha egy adott feladatot elvégzett.

### 2.3. Napi Teendők Megjelenítése

Napi teendők megjelenítése: A fő képernyőn minden naphoz (Hétfő, Kedd, Szerda, stb.) külön-külön képernyők tartoznak, ahol az adott napra mentett teendők és feladatok jelennek meg. A felhasználók böngészhetnek a különböző napokon és megtekinthetik azokhoz tartozó leírásokat. Teendők listázása és szűrése: A napi teendők a hozzájuk rendelt nap szerint kerülnek szűrésre és listázásra. Ha egy naphoz nincs teendő, akkor a képernyőn a "No notifications for this day" üzenet jelenik meg.

### 2.4. Navigáció

Navigáció a különböző képernyők között: Az alkalmazás lehetővé teszi a felhasználók számára, hogy a különböző képernyők között navigáljanak. A felhasználók beállíthatják a napi teendőket, valamint megtekinthetik azokat a napi nézetekben. Az alkalmazás alapértelmezetten a kezdőképernyővel (bejelentkezés vagy regisztráció) indul, majd sikeres bejelentkezés után a fő képernyőre, a napi teendők és profil menedzselésére van lehetőség. Tab navigáció: Az alkalmazás alsó navigációs sávja lehetővé teszi a felhasználók számára, hogy könnyedén váltsanak a napi teendők, az új teendők hozzáadása, és a profil kezelése között.

### 2.5. Profil Kezelés

Profil információk módosítása: A felhasználók módosíthatják profiljukat, például a jelszó megváltoztatásával. A profilkezelés során a felhasználó egy új jelszót adhat meg, amit az alkalmazás az Firebase backend-en keresztül módosít. Jelszó visszaállítása: A felhasználók képesek a jelszavukat visszaállítani, ha elfelejtették azt. A rendszer e-mailben küldhet visszaállítási linket.

### 2.6. Adatmegőrzés és Szinkronizálás

Teendők adatainak megőrzése: Az alkalmazás képes eltárolni a felhasználók által megadott napi teendőket, így azok a következő bejelentkezéskor is elérhetők maradnak. A teendőket az alkalmazás nem csak a helyi memória, hanem a Firebase Firestore adatbázis segítségével is tárolhatja. Szinkronizálás Firebase-el: A Firebase adatbázis és az autentikáció lehetővé teszi, hogy a felhasználói adatok és teendők szinkronizálva legyenek több eszköz között.

# 3. Használt Fejlesztési és Tesztelési Környezetek

## 3.1. Fejlesztési Környezet: Visual Studio Code

A fejlesztés során a **Visual Studio Code (VS Code)** volt az alkalmazott integrált fejlesztői környezet (IDE). A VS Code egy rendkívül népszerű, könnyű és rugalmas kódszerkesztő, amely számos hasznos funkcióval rendelkezik, amelyek megkönnyítik a React Native fejlesztést.

* **Kódszerkesztés**: A VS Code támogatja a JavaScript, JSX és React Native kódot, amely lehetővé teszi a hatékony kódszerkesztést, kódfelajánlásokat, hibák automatikus felismerését és szintaxis kiemelést.
* **Bővítmények**: A fejlesztés során különböző bővítményeket használtunk, mint például a **Prettier**, a **ESLint** és a **React Native Tools**. Ezek a bővítmények segítettek a kód formázásában, a szintaktikai hibák elkerülésében és a React Native projektek kezelésében.
* **Git Integráció**: A Visual Studio Code beépített Git integrációval rendelkezik, amely lehetővé tette a verziókezelést és a csapatmunka során a könnyed kódkezelést. Az IDE közvetlenül integrálható a GitHub-bal vagy más Git szolgáltatókkal.
* **Terminál**: A VS Code beépített terminálja segít a parancssori eszközök futtatásában, például a React Native CLI parancsokkal való interakcióban (például npx react-native run-android vagy npx react-native run-ios).

A Visual Studio Code használatának előnyei közé tartozik az egyszerűség, a gyors indítás, a testreszabhatóság és a rengeteg bővítmény, amelyek elősegítik a fejlesztési munkát, különösen a React Native alapú alkalmazások esetében.

## 3.2. Tesztelési Környezet: Android Studio

A tesztelés során az alkalmazás **Android Studio** környezetét használtuk, amely a Google által fejlesztett hivatalos integrált fejlesztői környezet (IDE) az Android alkalmazások fejlesztésére. Az Android Studio robusztus tesztelési eszköztárat biztosít, amely segíti az alkalmazások hibakeresését és optimalizálását Android platformon.

* **Android Emulator**: Az Android Studio emulátora lehetővé tette az alkalmazás tesztelését különböző Android verziókon és eszköztípusokon. A fejlesztés során különböző Android verziók és képernyőfelbontások konfigurálása segített a különböző hardverekre és szoftverekre történő optimalizálásban. Az emulátor különböző szcenáriók, például hálózati kapcsolati problémák, helymeghatározás és érzékeny interakciók tesztelésére is alkalmas.
* **Firebase Integration Testing**: Mivel az alkalmazás Firebase-t használ autentikációhoz és adatkezeléshez, az Android Studio segítségével teszteltük a Firebase funkciókat, beleértve a felhasználói regisztrációt, bejelentkezést és az adatbázis műveleteket.

Az Android Studio tesztelési környezetének alkalmazása biztosította, hogy az alkalmazás teljesítménye, funkcionálisan helyes működése, valamint felhasználói élménye minden környezetben optimális legyen.

# 4. Kód

## 4.1 Firebase Konfiguráció

A képen szöveg, elektronika, képernyőkép, képernyő látható

Automatikusan generált leírásA Firebase konfigurációját a firebaseConfig objektum tartalmazza, amely inicializálja az autentikációt (auth) és az adatbázist (db). A Firebase funkciók, mint a regisztráció, bejelentkezés, adatmentés, és lekérdezés központi szerepet töltenek be.

4.2 Navigáció

### 4.2.1 Main Nav.

Az alkalmazás két fő navigációs struktúrával rendelkezik: Stack Navigator: Az AuthScreen (kezdőlap) és a MainTabs között vált.

### 4.2.2 Bottom Nav.

Bottom Tab Navigator: A napokhoz kapcsolódó képernyők és a profilkezelés érhetők el.

A képen szöveg, képernyőkép, képernyő, szoftver látható

Automatikusan generált leírás

4.3 Komponensek

4.3.1 AuthScreen: Bejelentkezés és regisztráció kezelése.

4.3.2 DataSet: Napokhoz tartozó leírások hozzáadása.

4.3.3 ProfileScreen: Jelszó módosítása és kijelentkezés.

4.3.4 DayScreen: Napi értesítések megjelenítése.

4.3.5 MainTabs: A Tab.Navigator segítségével biztosítja a napok és profil közötti navigációt.

4.4. Funkciók Részletezése

4.4.1 Felhasználói Hitelesítés: Az AuthScreen két fő műveletet támogat: Regisztráció (handleSignUp): A Firebase createUserWithEmailAndPassword metódusával hoz létre új fiókot.

Bejelentkezés (handleLogin): A Firebase signInWithEmailAndPassword metódusával hitelesíti a felhasználót.

4.4.2 Adatkezelés és Mentés:

A DataSet komponens lehetővé teszi, hogy a felhasználók napokhoz kötött leírásokat adjanak hozzá. Az adatok mentése során: A bejegyzések a bookingDetails objektumba kerülnek. A fő alkalmazás állapotában (notifications) tárolódnak.

4.4.3 Napi Értesítések:

-A DayScreen az adott naphoz tartozó értesítéseket jeleníti meg, szűrve azokat a nap neve alapján. Profilkezelés

-A ProfileScreen jelszóváltoztatást és kijelentkezést valósít meg: Jelszó módosítása (handleChangePassword): A Firebase updatePassword funkcióját használja. Kijelentkezés (handleLogout): A Firebase signOut metódusa felelős az aktuális munkamenet lezárásáért.

A képen szöveg, elektronika, képernyőkép, képernyő látható

Automatikusan generált leírás

4.5 Stílusok

A StyleSheet definiálja az alkalmazás megjelenését.

A képen szöveg, képernyőkép, szoftver, Operációs rendszer látható

Automatikusan generált leírásA főbb elemek:

-Háttérkép: Az ImageBackground biztosít egységes hátteret.

-Űrlapok és Gombok: A styles.input és a styles.button az interaktív elemek egységes megjelenését biztosítják. Színek: Sötét színvilág (#333, #442cfc) dominál, fehér szövegekkel.